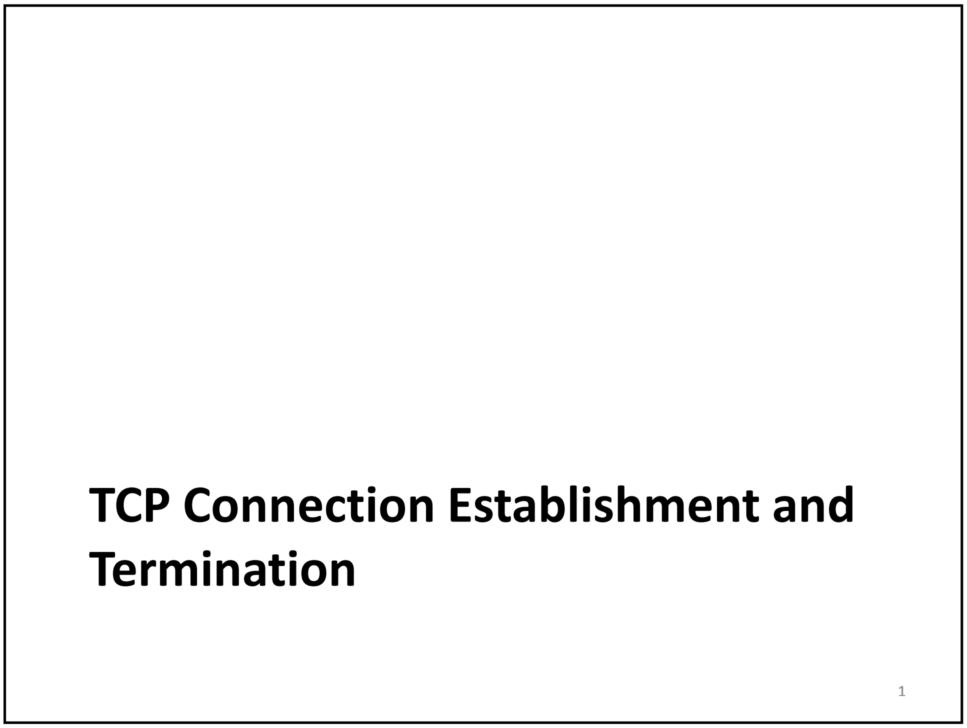
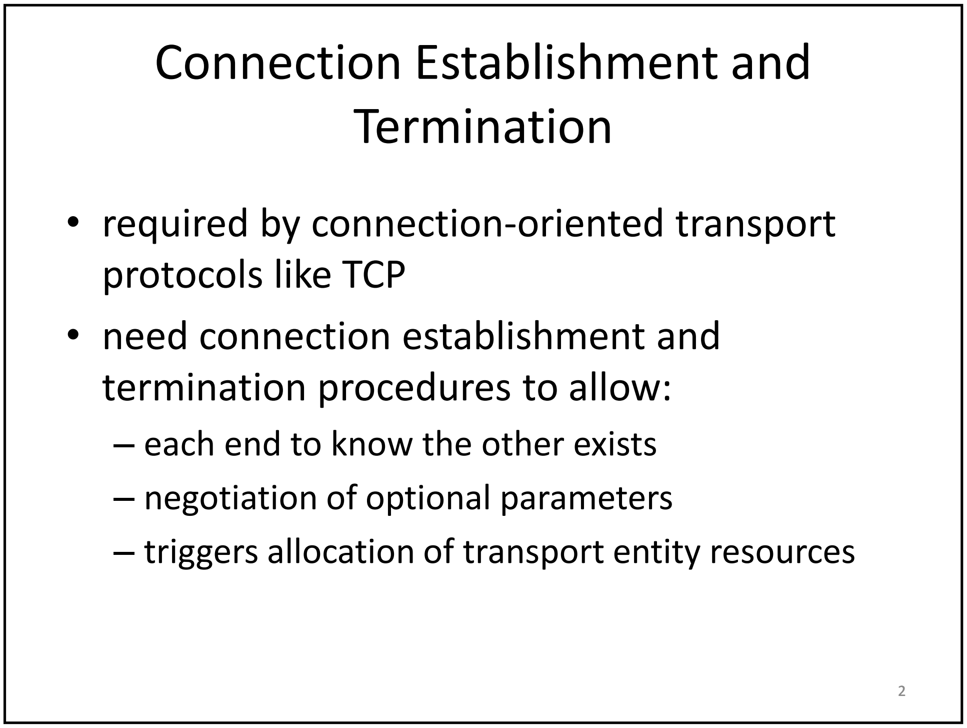


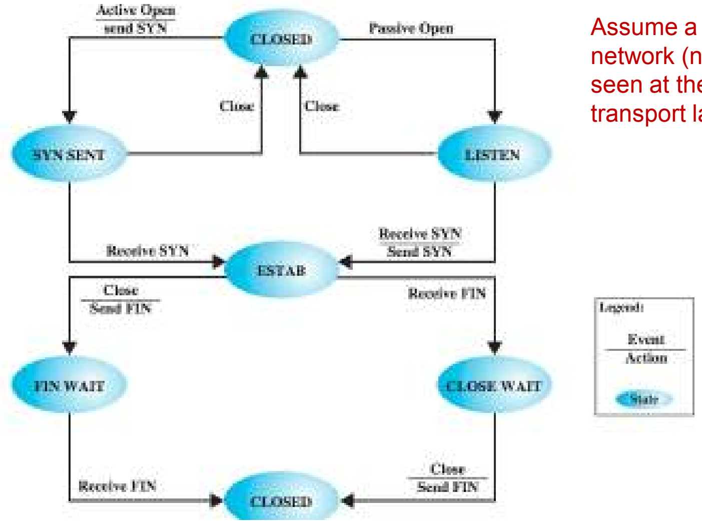
TCP Connection Establishment and Termination



Connection Establishment and Termination

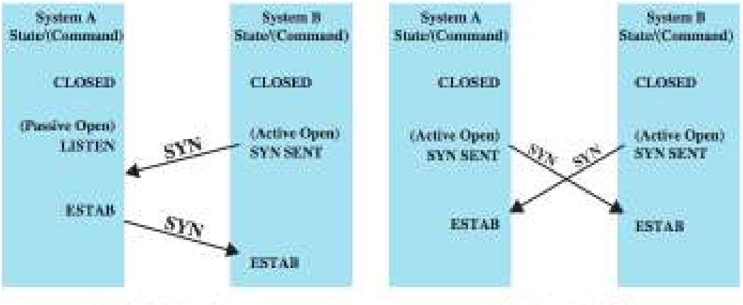
- required by connection-oriented transport protocols like TCP
- need connection establishment and termination procedures to allow:
  - each end to know the other exists
  - negotiation of optional parameters
  - triggers allocation of transport entity resources

Connection State Diagram



Assume a reliable network (no loss seen at the transport layer).

Connection Establishment Diagram



(a) Active/Passive Open

(b) Active/Active Open

Assume a reliable network (no loss seen at the transport layer).

What if either SYN is lost? (discussed later)

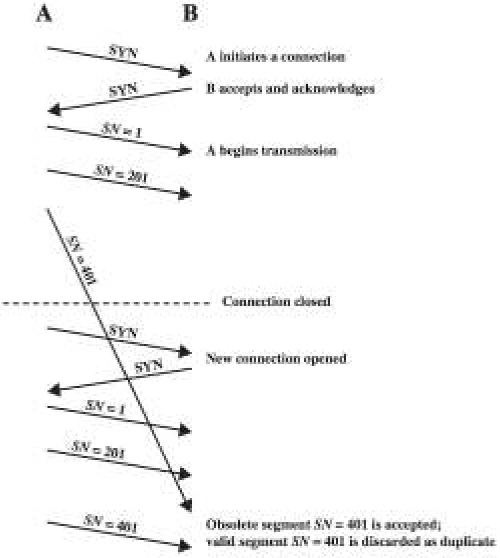
Connection Termination

- either or both sides by mutual agreement
- graceful or abrupt termination
- if graceful, initiator must:
  - send FIN to other end, requesting termination
  - place connection in FIN WAIT state
  - when FIN received, inform user and close connection
- other end must:
  - when receives FIN must inform TS user and place connection in CLOSE WAIT state
  - when TS user issues CLOSE primitive, send FIN & close connection

Connection Establishment

- two way handshake
  - A send SYN, B replies with SYN
  - lost SYN handled by re-transmission
  - ignore duplicate SYNs once connected
- lost or delayed data segments can cause connection problems
  - eg. segment from old connection

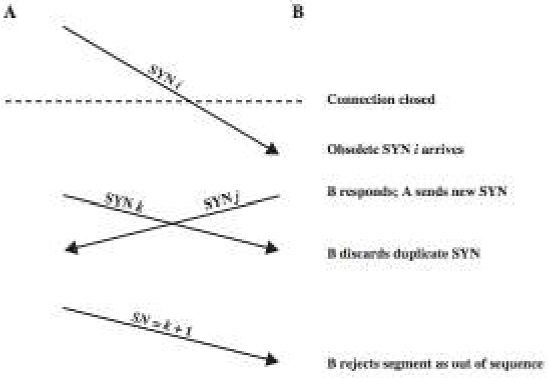
Two Way Handshake: Obsolete Data Segment



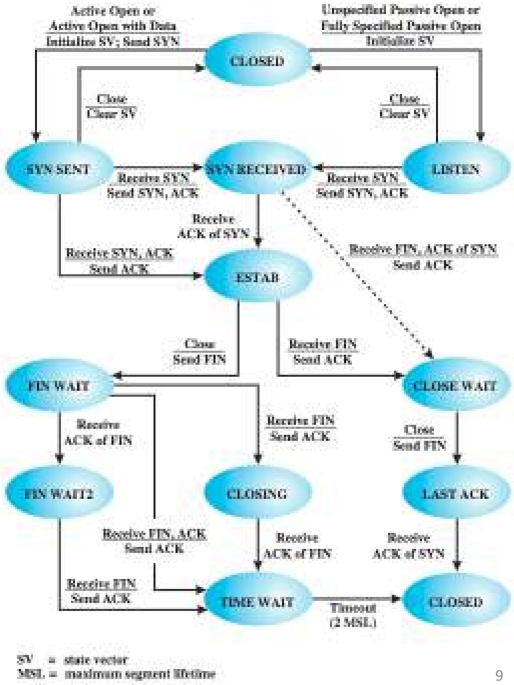
Solution: starting SN is far away from the last SN of the previous connection.

Use request of the form SYN***i*** where *i* +1 is the SN of the first data segment to be sent.

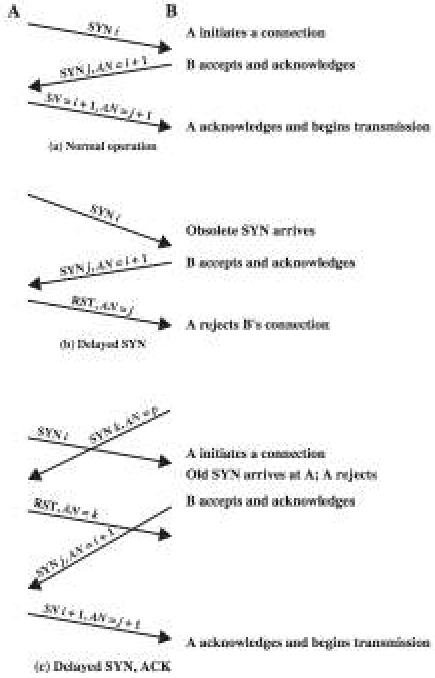
Two Way Handshake: Obsolete SYN Segment



TCP Three Way Handshake: State Diagram



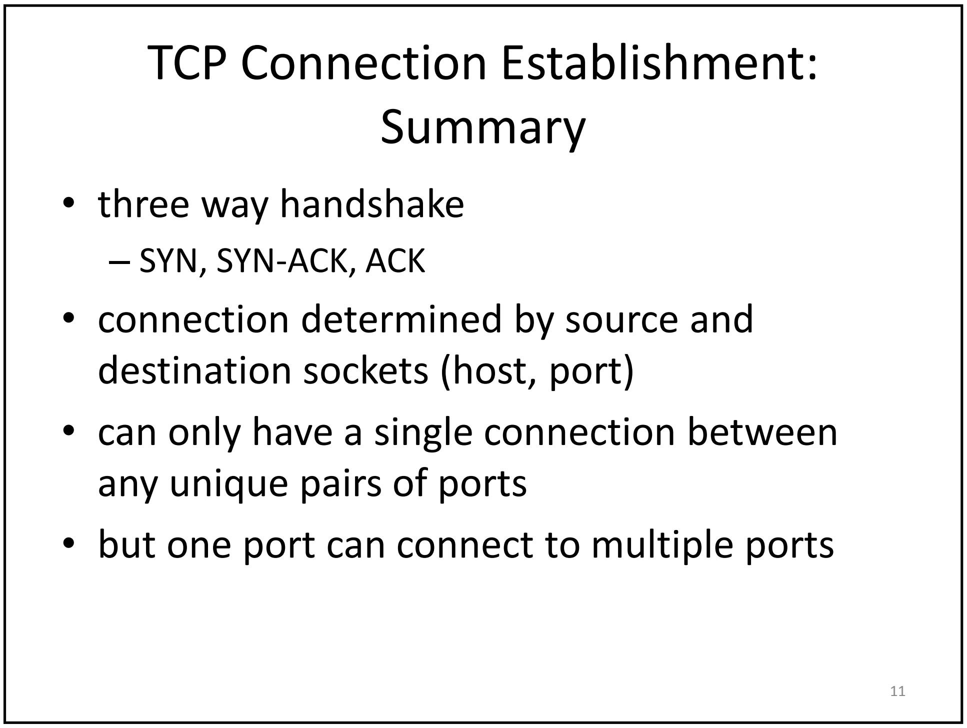
TCP Three Way Handshake: Examples



(a) Normal operation

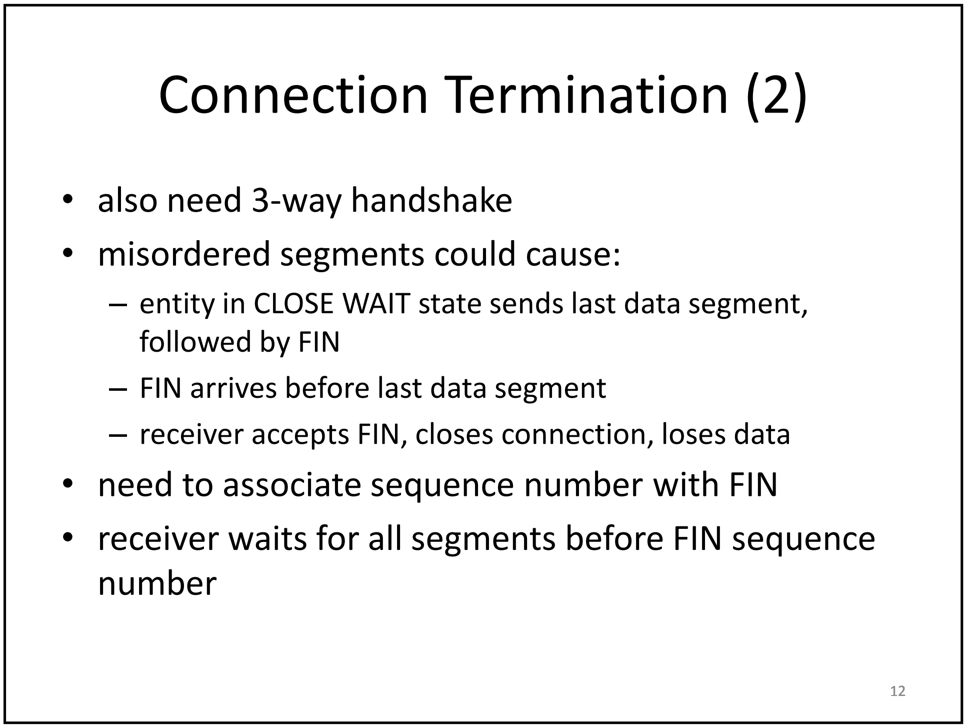
(b) Delayed SYN

(c) Delayed SYN, ACK



TCP Connection Establishment: Summary

- three way handshake
  - SYN, SYN-ACK, ACK
- connection determined by source and destination sockets (host, port)
- can only have a single connection between any unique pairs of ports
- but one port can connect to multiple ports



Connection Termination (2)

- also need 3-way handshake
- misordered segments could cause:
  - entity in CLOSE WAIT state sends last data segment, followed by FIN
  - FIN arrives before last data segment
  - receiver accepts FIN, closes connection, loses data
- need to associate sequence number with FIN
- receiver waits for all segments before FIN sequence number

Connection Termination: Graceful Close

- also have problems with loss of segments and obsolete segments
- need graceful close which will:
- send FIN i and receive AN i+1
- receive FIN j and send AN j+1
- wait twice maximum expected segment lifetime

Reading

- Chapter 20, Stallings’ book